

1. 安装VMware

2. 使用VMware安装一台Centos7操作系统

3. 关于CentOS7的一些必备配置

- 3.1. 安装一些必备的软件
- 3.2. 修改主机名
- 3.3. 配置IP
- 3.4. 添加主机域名ip映射
- 3.5. 设置系统默认启动级别
- 3.6. 关闭防火墙/关闭Selinux
 - 3.6.1. 防火墙操作相关
 - 3.6.2. 关闭Selinux
- 3.7. 关闭 THP
- 3.8. umask设置
- 3.9. 文件描述符配置
- 3.10. 同步服务器时间
- 3.11. 配置YUM源
- 3.12. 安装DK
- 3.13. 安装Maven
- 3.14. 安装Scala
- 3.15. 安装Python3
- 3.16. 安装MySQL

4. 准备一个集群

- 4.1. 执行虚拟机克隆
- 4.2. 配置SSH免密登录
 - 4.2.1. 第一种
 - 4.2.2. 第二种

5. 背景

6. 注意事项

7. 配置 Ambari本地YUM源

- 7.1. 下载安装ambari所需的一些资源
- 7.2. 安装配置 httpd 服务
- 7.3. 配置 YUM 源
- 7.4. 复制 MySQL 驱动
- 7.5. 安装 Amabari Server
- 7.6. 初始化 ambari 元数据库
- 7.7. 配置 Amabari
- 7.8. 启动Ambari-Sever
- 7.9. 部署集群

1. 安装VMware

详情见相关 文档 或者 视频

2. 使用VMware安装一台Centos7操作系统

详情见相关 文档 或者 视频

3. 关于CentOS7的一些必备配置

3.1. 安装一些必备的软件

```
yum -y install net-tools
yum -y install ntpdate
yum -y install wget
yum -y install perl
yum -y install vim*
yum -y install lrzsz
yum -y install yum-utils
yum -y install ntp ntpdate
yum -y install autoconf
yum -y install automake
yum -y install cyrus-sasl-devel
yum -y install cyrus-sasl-gssapi
yum -y install cyrus-sasl-plain
yum -y install flex
yum -y install gcc
yum -y install gcc-c++
yum -y install gdb
yum -y install krb5-server
yum -y install krb5-workstation
yum -y install libtool
yum -y install make
yum -y install openssl-devel
yum -y install patch
yum -y install pkgconfig
yum -y install redhat-lsb-core
yum -y install rsync
yum -y install unzip
yum -y install vim-common
yum -y install which
yum install -y bind-utils
yum install -y psmisc
yum install -y libxslt
yum install -y zlib
yum install -y sqlite
yum install -y fuse
yum install -y fuse-libs
yum install -y redhat-lsb
yum install -y mod_ssl
```

3.2. 修改主机名

查看主机名称：

```
hostnamectl
```

使用root账号登录，然后执行命令：

```
vi /etc/hostname
```

或者如果配置了 bigdata 的 sudoer 权限，则在 bigdata 登录情况下使用命令：

```
sudo vi /etc/hostname
```

指定好内容之后保存退出即可：

```
bigdata01
```

或者不使用编辑配置文件的方式，使用命令也可以进行永久更改：

```
hostnamectl set-hostname bigdata01
```

3.3. 配置IP

Linux 服务器的IP修改方式有三种，在此不细讲，请参考之前的学习文档。这里提供简易步骤：

第一步：进入到ip配置文件目录中

```
cd /etc/sysconfig/network-scripts
```

第二步：编辑IP配置文件

```
vim ifcfg-ens33
```

按需修改或者增加如下内容：

```
BOOTPROTO="static"  
ONBOOT="yes"  
IPADDR="192.168.123.151"  
GATEWAY="192.168.123.2"  
DNS1="192.168.123.2"
```

完整的如下：

```
TYPE="Ethernet"  
PROXY_METHOD="none"  
BROWSER_ONLY="no"  
BOOTPROTO="static"  
DEFROUTE="yes"
```

```
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens33"
UUID="c08cc7b8-0b9a-46d0-9fc5-f94a233ffa7e"
DEVICE="ens33"
ONBOOT="yes"
IPADDR="192.168.123.151"
PREFIX="24"
GATEWAY="192.168.123.2"
DNS1="192.168.123.2"
IPV6_PRIVACY="no"
```

第三步：重启网络服务

```
systemctl restart network
```

3.4. 添加主机域名ip映射

修改配置文件：

```
vi /etc/hosts
```

增加对应的内容：

```
192.168.123.101 bigdata01
192.168.123.102 bigdata02
192.168.123.103 bigdata03
192.168.123.104 bigdata04
192.168.123.105 bigdata05
192.168.123.106 bigdata06
192.168.123.107 bigdata07
192.168.123.108 bigdata08
192.168.123.109 bigdata09
192.168.123.110 bigdata10
192.168.123.111 bigdata11
192.168.123.112 bigdata12
192.168.123.113 bigdata13
192.168.123.114 bigdata14
192.168.123.115 bigdata15
192.168.123.116 bigdata16
192.168.123.121 bigdata21
192.168.123.122 bigdata22
192.168.123.123 bigdata23
192.168.123.124 bigdata24
192.168.123.125 bigdata25
192.168.123.126 bigdata26
192.168.123.131 bigdata31
192.168.123.132 bigdata32
192.168.123.133 bigdata33
```

```
192.168.123.134 bigdata34
192.168.123.135 bigdata35
192.168.123.136 bigdata36
```

修改好之后，保存退出即可。

3.5. 设置系统默认启动级别

这一步不是必须的，但是考虑到咱们用的是虚拟机，使用了桌面系统会占用相当多内存，而且在企业生产环境中，也都不会开启桌面系统，都是通过远程连接工具去链接使用。所以咱们最好还是要改一改。但是好在我们现在使用的是centos7，我们安装的如果是centos7的minimal系统的话，可以不用做任何更改的。centos7的minimal系统的默认级别就是multi-user.target，和centos6的level 3是一样的意思。

如果需要更改：

```
systemctl set-default multi-user.target
```

3.6. 关闭防火墙/关闭Selinux

注意：

生产环境一般由专业运维人员来限制相关端口，而不是完全禁用防火墙。
SELinux是美国国家宇航局开发的Linux的安全子系统，一定要禁用掉免得出现各种莫名其妙的问题。

3.6.1. 防火墙操作相关

关闭防火墙： 如果看到inactive(dead)就意味着防火墙关闭了

```
systemctl stop firewalld
```

永久关闭防火墙：

```
systemctl disable firewalld
```

3.6.2. 关闭Selinux

查看状态：

```
getenforce
```

临时关闭：

```
sudo setenforce 0
```

永久关闭，具体做法是修改 /etc/selinux/config 配置文件中的 SELINUX=disabled

```
vi /etc/selinux/config
```

修改如下配置项：

```
SELINUX=disabled
```

保存退出。

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3.7. 关闭 THP

如果不关闭THP，Hadoop的系统CPU使用率很高。

查看：

```
[root@bigdata15 ~]# cat /sys/kernel/mm/transparent_hugepage/defrag
[always] madvise never
[root@bigdata15 ~]# cat /sys/kernel/mm/transparent_hugepage/enabled
[always] madvise never
```

关闭：

```
vim /etc/rc.d/rc.local
```

在文末加入一段代码：

```
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
    echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
    echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
```

保存退出，然后赋予rc.local文件执行权限：

```
chmod +x /etc/rc.d/rc.local
```

重启生效。

3.8. umask设置

umask 用于设置在 Linux 上创建新文件或文件夹时授予的默认权限或基本权限。大多数 Linux 发行版将 022 设置为默认 umask 值。umask 值 022 授予新文件或文件夹的 755 权限。umask 值 027 授予新文件或文件夹的 750 权限。Ambari, HDP 和 HDF 支持的 umask 值为 022 (等价于 0022), 027 (等价于 0027)。这些值必须在所有主机上设置。查看当前:

```
umask
```

设置为 0022 即 022

```
sudo vi /etc/profile
```

在最后一行添加:

```
umask 0022
```

使环境变量生效:

```
source /etc/profile
```

再次用umask命令检查:

```
umask
```

3.9. 文件描述符配置

Linux操作系统会对每个进程能打开的文件数进行限制(某用户下某进程), Hadoop生态系统的很多组件一般都会打开大量的文件, 因此要调大相关参数(生产环境必须调大, 学习环境稍微大点就可以)。

检查文件描述符限制数, 可以用如下命令检查当前用户下一个进程能打开的文件数:

```
ulimit -Sn  
ulimit -Hn
```

结果:

```
1024  
4096
```

注意: 上面的命令只能检查当前用户bigdata。

建议的最大打开文件描述符数为10000或更多。要检查为最大打开文件描述符数设置的当前值, 请在每个主机上执行以下shell命令:


```
ulimit -Sn
ulimit -Hn
```

如果输出不大于10000，请运行以下命令将其设置为合适的默认值：

```
ulimit -n 10000
```

执行命令修改配置文件也可以：

```
vim /etc/security/limits.conf
```

直接内容：

```
# End of file
* soft nfile 65536
* hard nfile 65536
* soft nproc 131072
* hard nproc 131072
```

3.10. 同步服务器时间

对于一个Hadoop集群来说，每个节点的系统时间，应该要都是一样的。进行时间同步，可以有两种简单的方式，一是手动同步，而是搭建时间服务器进行同步或者从网络定时同步系统时间。推荐最后一种：从网络定时同步时间。

1、使用date命令手动简单同步一下

```
date -s "2016-10-23 11:11:11"
```

修改时间后，需要写入硬件bios才能在重启之后依然生效

```
hwclock -w
```

2、配置 crontab 服务，用ntpdate 定时同步时间（推荐方式）

```
*/10 * * * * ntpdate ntp.sjtu.edu.cn
```

请先测试一下ntpdate命令是否可以使用，如果不可以，使用命令安装一下：

```
[root@bigdata02 ~]# ntpdate time.nist.gov
[root@bigdata02 ~]# ntpdate 202.120.2.101
[root@bigdata02 ~]# ntpdate ntp.sjtu.edu.cn
```

3、如果类似 202.120.2.101 这种网络时间服务器无法访问，那么请自行搭建时间服务器

4、当然还有一个非常重要的点：**不要忘记更改时区，然后再同步时间**

CentOS中时区是以文件形式存在，当前正在使用的时区文件位于/etc/localtime，其他时区文件则位于

```
/usr/share/zoneinfo
```


其中中国时区使用

```
/usr/share/zoneinfo/Asia/Shanghai
```

则，更改时区的话，直接使用如下命令就OK

```
cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

如果没有 Asia/Shanghai 时区文件，请执行 tzselect 命令按照引导去生成时区文件，生成好的时区文件就在 /usr/share/zoneinfo 目录下。

3.11. 配置YUM源

- 1、配置本地 yum 源
- 2、配置国内远程 yum 源

```
yum -y install wget
wget http://mirrors.aliyun.com/repo/Centos-7.repo
mv /root/Centos-7.repo /root/Centos-7-aliyun.repo
cp /root/Centos-7-aliyun.repo /etc/yum.repos.d/
yum repolist
```

这个步骤最好做一下，但是是非必须的。

3.12. 安装JDK

3.13. 安装Maven

3.14. 安装Scala

3.15. 安装Python3

3.16. 安装MySQL

4. 准备一个集群

4.1. 执行虚拟机克隆

以上的步骤，一般来说，都是在一台机器中操作的。现在我们有两种方式进行集群节点的准备

第一种：

比如集群规划是4个节点，那么可以先安装好4个centos7系统，然后每个节点，都按照以上的步骤操作一遍。稍显麻烦。

第二种：

先准备一个节点，然后按照以上步骤操作完毕，然后执行虚拟机克隆。相对来说，速度要快很多。

具体执行虚拟机克隆，请参考相关的文档。

切记：恐龙完毕需要做几件事：

修改mac地址
修改主机名
修改IP地址

4.2. 配置SSH免密登录

对于一个集群来说，节点间的互信是必不可少的。所以一定要给集群的所有节点之间配置免密SSH访问。切记，你使用哪个用户搭建Hadoop集群，就给哪个用户配置SSH免密访问。另外需要注意的：所有节点的安装Hadoop的用户都一样该一样。我使用的就是 root 用户。

4.2.1. 第一种

在 root 用户登录状态下，执行命令生成公钥私钥对

```
ssh-keygen  
ssh-keygen -t rsa # 或者执行这个命令 -t 表示指定秘钥算法
```

之后你会发现，在/root/.ssh目录下生成了公钥文件

```
id_rsa      # 私钥  
id_rsa.pub  # 公钥
```

复制公钥文件到授权列表

```
cat ./id_rsa.pub >> authorized_keys
```

修改文件权限

```
chmod 600 ./authorized_keys
```

将该授权文件authorized_keys文件复制到slave节点

```
scp ./authorized_keys root@bigdata02:~/.ssh/
```

检查免密登录是否设置成功

```
ssh bigdata02
```

看看是不是能登录进入bigdata02的服务器，如果能登陆，则证明配置成功。

4.2.2. 第二种

在root用户登录状态下，执行命令生成公钥私钥对

```
ssh-keygen  
ssh-keygen -t rsa # 或者执行这个命令 -t 表示指定密钥算法
```

之后你会发现，在/root/.ssh目录下生成了公钥文件

```
id_rsa      # 私钥  
id_rsa.pub  # 公钥
```

使用一个更简单的方式，使用命令：

```
ssh-copy-id bigdata02 # 该命令的作用，就是把当前用户的公钥文件内容，发送到hadoop02这台  
机器的相同用户的授权文件中。  
ssh-copy-id bigdata03  
ssh-copy-id bigdata04  
ssh-copy-id bigdata05
```

建立bigdata01到bigdata02的免密登录，切记，我把公钥发送给对方，我就可以访问对方。而不是对方可以访问我。

检查免密登录是否设置成功

```
ssh bigdata02
```

看看是不是能登录进入bigdata02的服务器，如果能登陆，则证明配置成功。

5. 背景

虽然大数据越来越流行，但其学习的门槛却一直阻碍着很多的分布式应用初学者或者大数据的业务应用开发者。多个产品之间的不兼容问题，快速集成和维护也显得比较困难。不管是 Hadoop V1 或者 V2 的安装，又或者 Spark/YARN 等的集成，都不是几行简单的命令可以完成的，而是需要手工修改很多的集群配置，这进一步增加了业务开发者的学习和使用难度。有了 Ambari，这些都不再是难题。

6. 注意事项

先明确几个概念：

- 1、Ambari只能安装Hortonworks Data Platform，即Hortonworks的开源Hadoop，不支持Apache的Hadoop平台；
- 2、对于已经安装了Apach Hadoop或者其他Hadoop平台的，不能使用Ambari来管理；

再说几个注意事项：

- 1、Ambari默认的安装方式是使用yum，从远程下载HDP组件安装，而HDP平台安装包都非常大（本例中使用的HDP-2.4.0安装包为6G），Ambari平台有又30分钟的Timeout限制，如果在30分钟内下载不完HDP，就会造成安装失败。建议修改下载源文件，配置为本地源；
- 2、Ambari安装过程为自动安装，自动安装脚本会创建很多用户和其他组件（如系统自带的Java和数据库），建议使用一套干净的环境来安装。
- 3、如第2点所示，脚本会自动创建用户和安装组件，建议使用root用户；
- 4、系统请关闭selinux、防火墙和THP；
- 5、集群机器请事先配置ssh互信；还是建议使用root用户来互信；

7. 配置 Ambari本地YUM源

7.1. 下载安装ambari所需的一些资源

访问Cloudera的官网，进入到如下网址界面：https://docs.cloudera.com/HDPDocuments/Ambari-2.7.5.0/bk_ambari-installation/content/ambari_repositories.html

请下载以下的资源包：

Ambari相关安装包： <http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.7.4.0/ambari-2.7.4.0-centos7.tar.gz>

HDP安装： <http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.4.0/HDP-3.1.4.0-centos7-rpm.tar.gz>

HDP-UTIL安装包： <http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/centos7/HDP-UTILS-1.1.0.22-centos7.tar.gz>

HDP-GPL文件： <http://public-repo-1.hortonworks.com/HDP-GPL/centos7/3.x/updates/3.1.4.0/HDP-GPL-3.1.4.0-centos7-gpl.tar.gz>

考虑大下载速度很慢，老师也提供了离线包，可以直接使用。在百度云网盘。

7.2. 安装配置 httpd 服务

```
yum -y install httpd
systemctl start httpd
systemctl status httpd
systemctl enable httpd
```

7.3. 配置 YUM 源

先上传这四个文件 到 http 服务中：

```
ambari-2.7.4.0-centos7.tar.gz
HDP-3.1.4.0-centos7-rpm.tar.gz
HDP-GPL-3.1.4.0-centos7-gpl.tar.gz
HDP-UTILS-1.1.0.22-centos7.tar.gz
```

然后解压：

```
mkdir -p /var/www/html/hdp314-centos7
tar -zxvf ~/soft/ambari-2.7.4.0-centos7.tar.gz -C /var/www/html/hdp314-centos7
tar -zxvf ~/soft/HDP-3.1.4.0-centos7-rpm.tar.gz -C /var/www/html/hdp314-centos7
tar -zxvf ~/soft/HDP-GPL-3.1.4.0-centos7-gpl.tar.gz -C /var/www/html/hdp314-centos7
tar -zxvf ~/soft/HDP-UTILS-1.1.0.22-centos7.tar.gz -C /var/www/html/hdp314-centos7
```

观摩一下：

```
cd /var/www/html/
```

路径分别为：

```
/var/www/html/hdp314-centos7/HDP/centos7/3.1.4.0-315
/var/www/html/hdp314-centos7/HDP-GPL/centos7/3.1.4.0-315
/var/www/html/hdp314-centos7/HDP-UTILS/centos7/1.1.0.22
/var/www/html/hdp314-centos7/ambari/centos7/2.7.4.0-118
```

配置 yum 源：

```
cd /etc/yum.repos.d/
vim ambari.repo
```

注意上述这个 repo 文件名只能叫做：ambari.repo，否则后续使用过程中，会报错。

```
[CentOS7-HDP]
name=CentOS7-HDP
baseurl=http://bigdata22/hdp314-centos7/HDP/centos7/3.1.4.0-315
gpgcheck=0
enabled=1

[CentOS7-HDP-GPL]
name=CentOS7-HDP-GPL
baseurl=http://bigdata22/hdp314-centos7/HDP-GPL/centos7/3.1.4.0-315
gpgcheck=0
enabled=1

[CentOS7-HDP-UTILS]
name=CentOS7-HDP-UTILS
baseurl=http://bigdata22/hdp314-centos7/HDP-UTILS/centos7/1.1.0.22
gpgcheck=0
enabled=1

[CentOS7-ambari]
name=CentOS7-ambari
baseurl=http://bigdata22/hdp314-centos7/ambari/centos7/2.7.4.0-118
gpgcheck=0
enabled=1
```

生效：

清除缓存：

```
yum clean all
```

生成缓存:

```
yum makecache
```

重新检测:

```
yum repolist
```

7.4. 复制 MySQL 驱动

复制 MySQL JDBC 驱动 Jar 包到 /usr/share/java/ 目录

```
mkdir -p /usr/share/java/  
cd /usr/share/java/
```

```
cp ~/soft/mysql-connector-java-5.1.45-bin.jar /usr/share/java/
```

并且需要改名:

```
mv /usr/share/java/mysql-connector-java-5.1.45-bin.jar /usr/share/java/mysql-  
connector-java.jar
```

7.5. 安装 Amabari Server

执行命令:

```
yum -y install ambari-server
```

稍等片刻, ambari-server就安装好了

7.6. 初始化 ambari 元数据库

进入 MySQL 数据库命令行:

```
mysql -uroot -pQwer_1234
```

执行如下一些命令:


```

set global validate_password_policy=0;
set global validate_password_mixed_case_count=0;
set global validate_password_number_count=3;
set global validate_password_special_char_count=0;
set global validate_password_length=3;
create database ambari DEFAULT CHARSET utf8 COLLATE utf8_general_ci;
use ambari;
source /var/lib/ambari-server/resources/Ambari-DDL-MySQL-CREATE.sql;
exit

```

```

CREATE USER 'ambari'@'%' IDENTIFIED BY 'ambari';
GRANT ALL ON ambari.* TO 'ambari'@'%';
FLUSH PRIVILEGES;

```

7.7. 配置 Amabari

执行如下命令开始设置Ambari-Server:

```
ambari-server setup
```

```

[root@bigdata22 ~]# ambari-server setup
Using python /usr/bin/python
Setup ambari-server
Checking SELinux...
SELinux status is 'disabled'
Customize user account for ambari-server daemon [y/n] (n)?

```

是否自定义ambari-server的运行用户: 我们选择 N

```

Customize user account for ambari-server daemon [y/n] (n): n
Adjusting ambari-server permissions and ownership...
Checking firewall status...
Checking JDK...
[1] Oracle JDK 1.8 + Java Cryptography Extension (JCE) Policy Files 8
[2] Custom JDK
=====
Enter choice (1): 2
WARNING: JDK must be installed on all hosts and JAVA_HOME must be valid on all hosts.
WARNING: JCE Policy files are required for configuring Kerberos security. If you plan to u
sdiction Policy Files are valid on all hosts.
Path to JAVA_HOME: /usr/local/java/jdk1.8.0_181

```

我们选择2, 代表自定义java, 然后回车: 然后指定自己的JDK的安装路径。然后回车

```

Path to JAVA_HOME: /usr/local/java/jdk1.8.0_181
Validating JDK on Ambari Server...done.
Check JDK version for Ambari Server...
JDK version found: 8
Minimum JDK version is 8 for Ambari. Skipping to setup different JDK for Ambari Server.
Checking GPL software agreement...
GPL License for LZ0: https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html
Enable Ambari Server to download and install GPL Licensed LZ0 packages [y/n] (n)? y

```

提示是否下载并安装LZO的包, 我们输入y并回车: 是否进入数据库高级设置, 我们输入y然后回车:


```

Enable Ambari Server to download and install GPL Licensed LZO packages [y/n] (n)? y
Completing setup...
Configuring database...
Enter advanced database configuration [y/n] (n)? y
Configuring database...
=====
Choose one of the following options:
[1] - PostgreSQL (Embedded)
[2] - Oracle
[3] - MySQL / MariaDB
[4] - PostgreSQL
[5] - Microsoft SQL Server (Tech Preview)
[6] - SQL Anywhere
[7] - BDB
=====
Enter choice (1): 3

```

依次按照提示，输入你的数据库的相关信息：

```

Enter advanced database configuration [y/n] (n)? y
Configuring database...
=====
Choose one of the following options:
[1] - PostgreSQL (Embedded)
[2] - Oracle
[3] - MySQL / MariaDB
[4] - PostgreSQL
[5] - Microsoft SQL Server (Tech Preview)
[6] - SQL Anywhere
[7] - BDB
=====
Enter choice (1): 3
Hostname (localhost): bigdata22
Port (3306): 3306
Database name (ambari): ambari
Username (ambari): root
Enter Database Password (bigdata):
Re-enter password:
Configuring ambari database...
Should ambari use existing default jdbc /usr/share/java/mysql-connector-java.jar [y/n] (y)? y
Configuring remote database connection properties...

```

最后一步输入 y 回车：

```

Should ambari use existing default jdbc /usr/share/java/mysql-connector-java.jar [y/n] (y)? y
Configuring remote database connection properties...
WARNING: Before starting Ambari Server, you must run the following DDL directly from the database:
/usr/resources/Ambari-DDL-MySQL-CREATE.sql
Proceed with configuring remote database connection properties [y/n] (y)? y
Extracting system views...
ambari-admin-2.7.4.0.118.jar
....
Ambari repo file doesn't contain latest json url, skipping repoinfos modification
Adjusting ambari-server permissions and ownership...
Ambari Server 'setup' completed successfully.
[root@bigdata22 ~]#

```

7.8. 启动Ambari-Sever

在 bigdata22 上启动 amabri-server:

```
ambari-server start
```

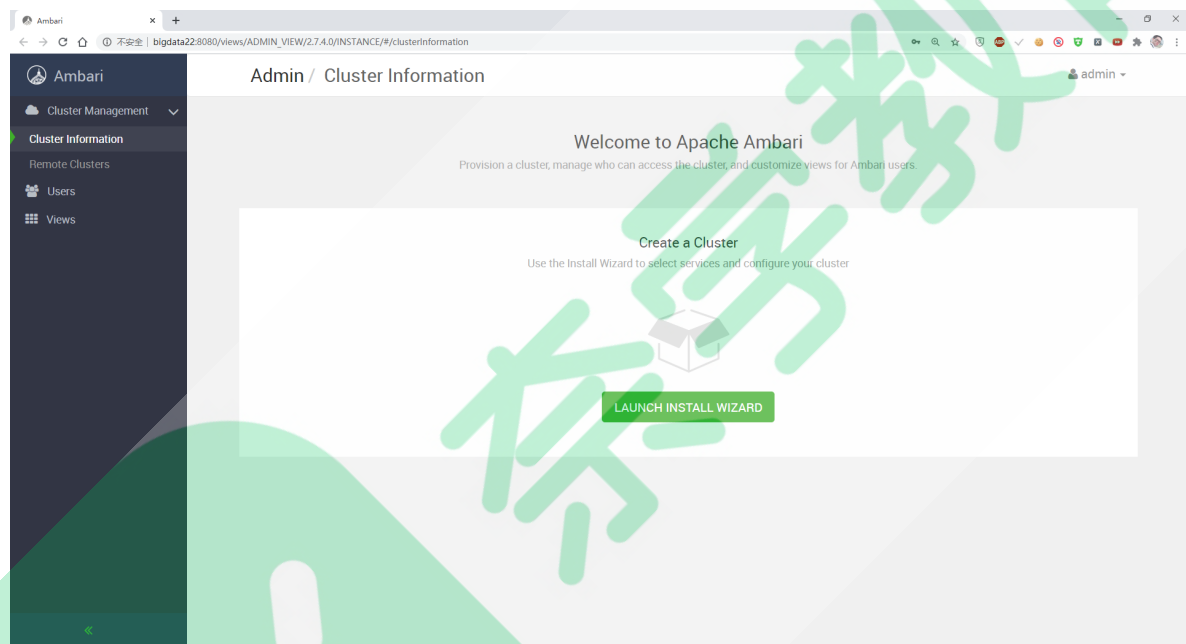
如果出现错误，一定要查看日志：

```
/var/log/ambari-server/ambari-server.log
```

出现如下界面就启动成功了：

```
[root@bigdata22 ~]# jps
2162 Jps
2020 AmbariServer
[root@bigdata22 ~]# netstat -nlt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1209/sshd
tcp6       0      0 :::8440                 :::*                     LISTEN      2020/java
tcp6       0      0 :::8441                 :::*                     LISTEN      2020/java
tcp6       0      0 :::443                  :::*                     LISTEN      1207/httpd
tcp6       0      0 :::3306                  :::*                     LISTEN      1295/mysqld
tcp6       0      0 :::8080                 :::*                     LISTEN      2020/java
tcp6       0      0 :::80                   :::*                     LISTEN      1207/httpd
tcp6       0      0 :::22                   :::*                     LISTEN      1209/sshd
[root@bigdata22 ~]#
```

访问 web UI: <http://bigdata22:8080>



7.9. 部署集群